AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) A register file for a data processing system comprising:

an unbanked memory unit having a plurality of registers addressable by an encoded address, wherein the encoded address corresponds to a respective one of the plurality of registers and a corresponding processor mode;

input ports to receive inputs for addressing at least one register using an encoded address, ~~and~~

output ports to output data from at least one register addressable by an encoded address; and

an address encoder, for each input port, the address encoder to provide an encoded address for accessing one of the plurality of registers.

2. (Original) The register file of claim 1, wherein the encoded address identifies a general purpose register associated with a processor mode.

3. (Original) The register file of claim 1, wherein each register is associated with a register index that maps to an encoded address based on at least one processor mode.

4. (Original) The register file of claim 3, wherein the input ports receive at least one source register index input and processor mode input for use in providing an encoded address for accessing at least one register.

5. (Cancelled)

6. (Currently Amended) The register file of claim [[5]] 3, further comprising:

a latch to latch an encoded address from the address encoder; and

a selector coupled to the latch and the address encoder, the selector to select the encoded address from either the latch or the address encoder.

7. (Original) The register file of claim 6, wherein the latch stores the encoded address as a pipeline storage of the encoded address.

8. (Previously Presented) The register file of claim 4, wherein data for one or more instructions being processed is outputted from the unbanked memory unit.

9. (Original) The register file of claim 3, further comprising:

input ports to receive at least one write index input and processor mode input for use in providing the encoded address for writing data to at least one register; and

at least one write input port for writing the data to the register addressable by the encoded address.

10. (Previously Presented) The register file of claim 9, wherein data for one or more executed instructions for the data processing are written into the unbanked memory unit.

11. (Original) The register file of claim 3, wherein the processor mode includes exception handling modes.

12. (Original) The register file of claim 11, wherein the exception handling processor modes include at least one of a fast interrupt request (FIQ) mode, interrupt request (IRQ) mode, supervisor (SVC) mode, undefined instruction (UND), and abort exception (ABT) mode. The register file of claim 12, wherein each exception handling triode corresponds to one or more registers.

13. (Original) The register file of claim 12, wherein each exception handling mode corresponds to one or more registers.

14. (Currently Amended) A register file for a data processing system comprising:

unbanked memory means having a plurality of register means addressable by an encoded address, wherein the encoded address corresponds to a respective one of the plurality of register means and a corresponding processor mode;

a plurality of input means for receiving inputs for addressing at least one register means using an encoded address; ~~and~~

output means for outputting data from at least one register means addressable by an encoded address; and

corresponding addressing means for each of the plurality of input means for providing an encoded address to address one of the plurality of register means.

15. (Original) The register file of claim 14, wherein the encoded address identifies a general purpose register associated with a processor mode.

16. (Original) The register file of claim 14, wherein each register means is associated with a register index that maps to an encoded address based on at least one processor mode.

17. (Original) The register file of claim 16, wherein the input means further includes means for receiving al least one source register index input and processor mode input for use in providing an encoded address for accessing at least register means.

18. (Cancelled)

19. (Currently Amended) The register file of claim [[18]] 17, further comprising:

~~a~~ latching means for latching an encoded address from the ~~address encoder~~ addressing means; and

a selecting means coupled to the latching means and the addressing means for selecting the encoded address from either the latching means or the addressing means.

20. (Original) The register file of claim 19, wherein the latching means includes storage means for storing the encoded address as a pipeline storage of the encoded address.

21. (Previously Presented) The register file of claim 17, wherein data for one or more instructions being processed is outputted from the unbanked memory means.

22. (Original) The register file of claim 16, further comprising:

input means for receiving at least one write index input and processor mode input for use in providing the encoded address for writing data to at least one register means; and

at least one write input means for writing the data to the register means addressable by the encoded address.

23. (Previously Presented) The register file of claim 22, wherein data for one or more executed instructions for the data processing are written into the unbanked memory means.

24. (Original) The register file of claim 16, wherein the processor mode includes exception handling modes.

25. (Original) The register file of claim 24, wherein the exception handling processor modes include at least one of a fast interrupt request (FIQ mode, interrupt request (rRQ) mode, supervisor (SVC) mode, undefined instruction (UND) mode, and abort exception (ABT) mode.

26. (Original) The register file of claim 25, wherein each exception handling mode corresponds to one or more register means.

27. (Currently Amended) A data processing system comprising: a microprocessor comprising:

a plurality of pipeline stages including a register file, the register file including:

an unbanked memory unit having a plurality of registers addressable by an encoded address, wherein the encoded address corresponds to a respective one of the plurality of registers and a corresponding processor mode;

input ports to receive inputs for addressing at least one register using an encoded address, wherein each of the input ports receives the inputs from a corresponding address encoder; and

output ports to output data from at least one register addressable by an encoded address.

28. (Original) The data processing system of claim 27, wherein the encoded address identifies a general purpose register associated with a processor mode.

29. (Previously Presented) The data processing system of claim 27, wherein the pipeline stages include:

an instruction fetch stage to fete one or more instructions; and

an instruction decode stage to decode fetched instructions from the instruction fetch stage, the instruction decode stage to forward inputs to the unbanked memory unit for outputting data from or writing data to one or more of the registers.

30. (Original) The data processing system of claim 29, wherein the register file further includes:

a plurality of input ports to receive inputs from the instruction decode stage, the inputs being used to obtain the encoded address for accessing at least one register; and

at least one output port to output data from the register addressable by the encoded address.

31. (Original) The data processing system of claim 30, wherein the pipeline stages further include:

an execution stage including a plurality of execution units, each execution unit to receive data from the register file for executing an instruction.

32. (Original) The data processing system of claim 31, further comprising:

a write back or retire logic stage to receive results data associated with one or more instructions executed by the execution units of the execution stage, and to forward the results data to the register file for storage.

33. (Original) The data processing system of claim 32, wherein the register file further includes:

a plurality of input ports to receive the data from the write back or retire logic for one or more executed instructions, the data to be written in the register file.

34. (Original) The data processing system of claim 29, wherein each register is associated with a register index that maps to one of the encoded addresses based on at least one processor mode.

35. (Original) The data processing system of claim 34, wherein the processor mode includes exception handling modes.

36. (Original) The data processing system 35, wherein the exception handling processor modes include at least one of a fast interrupt request (FIQ) mode, interrupt request (IRQ) mode, supervisor (SVC) mode, undefined instruction mode (UND), and abort exception (ABT) mode.

37. (Original) The data processing system of claim 36, wherein each exception handling mode corresponds to one or more registers.

38. (Currently Amended) A data processing system comprising:

a processing means for processing instructions,

a pipeline means for executing instructions, the pipeline means including a register file means, the register file means including:

unbanked memory means having a plurality of register means addressable by an encoded address, wherein the encoded address corresponds to a respective one of the plurality of register means and a corresponding processor mode,

a plurality of input means for receiving inputs for addressing at least one register means using an encoded address; ~~and~~

output means for outputting output data from at least one register means addressable by an encoded address; and

addressing means for each of the plurality of input means for providing an encoded address for accessing one of the plurality of register means.

39. (Original) The register file of claim 38, wherein the encoded address identifies a general purpose register associated with a processor mode.

40. (Previously Presented) The data processing system of claim 38, wherein the pipeline means includes:

a fetching means for fetching one or more instructions for execution; and

decoding means for decoding fetched instructions from the fetching means and for forwarding inputs to the unbanked memory means for outputting data from or writing data to one or more of the register means.

41. (Original) The data processing system of claim 40, wherein the register file means further includes:

input means for receiving inputs from the decoding means, the inputs being used to obtain the encoded address for accessing at least one register; and

at least one output means for outputting data from the register addressable by the encoded address.

42. (Original) The data processing system of claim 41, wherein the pipeline means further includes: an execution means including a plurality of execution processing means, each execution processing means receiving receive data from the register file means for executing an instruction.

43. (Original) The data processing system of claim 42, further comprising:

a write back means or retire logic means for receiving results data associated with one or more instructions executed by an execution processing means and for forwarding the results data to the register file means for storage.

44. (Original) The data processing system of claim 43, wherein the register file means further includes:

a plurality of input means for receiving the data from the write back means or retire logic means for one or more executed instructions, the data to be written in the register file means.

45. (Original) The data processing system of claim 40, wherein each register means is associated with a register index that maps to one of the encoded addresses based on al least one processor mode.

46. (Original) The data processing system of claim 45, wherein the processor mode includes exception handling modes.

47. (Original) The data processing system 46, wherein the exception handling processor modes include at least one of a fast interrupt request (FIQ mode, interrupt request (IRQ) mode, supervisor (SVC) mode, undefined instruction (LJND) mode, and abort exception (ABT) mode.

48. (Original) The data processing system of claim 47, wherein each exception handling mode corresponds to one or more register means.

49. (Currently Amended) A microprocessor comprising:

an integrated circuit comprising:

an unbanked memory unit having a plurality of registers addressable by an encoded address, wherein the encoded address corresponds to a respective one of the plurality of registers and a corresponding processor mode; ~~and~~

a plurality of inputs to receive index and processor mode information for use in providing the encoded address;

at least one output to output data stored in the storage location addressable by the encoded address; and

at least one address encoder for each of the inputs to provide at least one encoded address for addressing at least one of the registers.

50. (Original) The microprocessor of claim 49, wherein the encoded address identifies a general purpose register associated with a processor mode.

51. (Original) The microprocessor of claim of claim 49, wherein each register is associated with a register index that maps to the encoded address based on at least one processor mode.

52. (Original) The microprocessor of claim 51, wherein the processor mode includes exception handling modes.

53. (Original) The microprocessor of claim 52, wherein the exception handling processor modes include at least one of a fast interrupt request (FIQ mode, interrupt request (IRQ) mode, supervisor (SVC) mode, undefined instruction (UND) erode, and abort exception (ABT) mode.

54. (Cancelled)

55. (Currently Amended) The microprocessor of claim 51, further comprising:

~~at least one input to receive index and processor mode information for use in providing the encoded address; and~~

at least one write input to receive data to be written into the storage location addressable by the encoded address.

56. (Cancelled)

57. (Currently Amended) A microprocessor comprising:

an integrated circuit means comprising:

unbanked memory means having a plurality of register means addressable by an encoded address, wherein the encoded address corresponds to a respective one of the plurality of register means and a corresponding processor mode; ~~and~~

a plurality of input means for receiving index and processor mode information for use in providing the encoded address;

at least one output means for outputting data stored in the storage location addressable by the encoded address; and

at least one addressing means for each of the plurality of input means for providing at least one encoded address for addressing at least one of the register means.

58. (Original) The microprocessor of claim 57, wherein the encoded address identifies a general purpose register associated with a processor mode.

59. (Original) The microprocessor of claim of claim 57, wherein each register means is associated with a register index that maps to the encoded address based on at least one processor mode.

60. (Original) The microprocessor of claim 59, wherein the processor mode includes exception handling modes.

61. (Original) The microprocessor of claim 60, wherein the exception handling processor modes include at least one of a fast interrupt request (FIQ mode, interrupt request (IRQ) mode, supervisor (SVC) mode, undefined instruction (UND) mode, and abort exception (ABT) mode.

62. (Original) The microprocessor of claim 61, wherein the exception handling modes correspond to one or more registers.

63. (Cancelled)

64. (Currently Amended) The microprocessor of claim 59, further comprising:

~~at least one input means for receiving index and processor mode information for use in providing the encoded address; and~~

at least one write input means for receiving data to be written into the storage location addressable by the encoded address.

65. (Currently Amended) An integrated circuit method comprising:

configuring the integrated circuit to receive processor mode and source data inputs;

configuring the integrated circuit to determine an encoded address at one of a plurality of address encoders based on the processor mode and source data inputs, wherein the encoded address corresponds to a respective one of a plurality of unbanked registers and a corresponding processor mode and each of the processor mode and source data inputs corresponds to one of the plurality of address encoders;

configuring the integrated circuit to address one of the unbanked registers using an encoded address; ~~and~~

configuring the integrated circuit to output data from the unbanked register addressable by the encoded address.

66. (Original) The method of claim 65, further comprising:

configuring the integrated circuit to output data for multiple instructions.

67. (Previously Presented) The method of claim 65, further comprising:

configuring the integrated circuit to write data to one of the unbanked registers addressable by an encoded address.

68. (Previously Presented) The method of claim 67, further comprising:

configuring the integrated circuit to write data to one or more unbanked registers for multiple executed instructions.

69. (Currently Amended) A method for accessing an unbanked memory unit having a plurality of registers comprising:

receiving inputs for accessing the register file;

determining at least one encoded address in accordance with the received inputs at one of a plurality of address encoders, wherein at least one of the plurality of address encoders corresponds to each of the received inputs;

accessing the unbanked memory unit in accordance with the encoded address, wherein the encoded address corresponds to a respective one of the plurality of registers and a corresponding processor mode; and

outputting data from the unbanked memory unit accessed with the encoded address.

70. (Original) The method of claim 69, wherein receiving the inputs includes receiving processor mode inputs and source data inputs.

71. (Original) The method of claim 70, wherein determining at least one encoded address includes determining at least one encoded address based on the processor mode inputs and source data inputs.

72. (Original) The method of claim 69, wherein outputting data includes outputting data for multiple instructions.

73. (Original) The method of claim 69, further comprising:

writing data to at least one of the registers addressable by the encoded address.

74. (Original) The method of claim 73, further comprising:

writing write data to the registers for multiple executed instructions.

75. (New) The register file of claim 1, wherein each of the plurality of registers has an address having a length of x bits, each of the processor modes has a length of y bits, and the encoded address has a length that is less than x + y bits.